JOURNAL OF COMPUTATIONAL PHYSICS 107, 388–402 (1993)

Bounded Multi-scale Plasma Simulation: Application to Sheath Problems

S. E. PARKER*

Electronics Research Laboratory, University of California, Berkeley, California 94720

A. FRIEDMAN AND S. L. RAY

Lawrence Livermore National Laboratory, University of California, Livermore, California 94550

AND

C. K. BIRDSALL

Electronics Research Laboratory, University of California, Berkeley, California 94720

Received July 13, 1990; revised July 21, 1992

In our previous paper we introduced the *multi-scale* method, a self-consistent plasma simulation technique that allowed particles to have independent timesteps. Here we apply the method to one-dimensional electrostatic bounded plasma problems and demonstrate a significant reduction in computing time. We describe a technique to allow for variable grid spacing and develop consistent boundary conditions for the direct implicit method. Also discussed are criteria for specifying timestep size as a function of position in phase space. Next, an analytically solvable sheath problem is presented, and a comparison to simulation results is made. Finally, we show results for an ion acoustic shock front propagating toward a conducting wall. © 1993 Academic Press, Inc.

1. INTRODUCTION

Plasmas embody a wide range of space and time scales. For example, the important time scales for a collisionless and unmagnetized plasma can range from the electron plasma period to the ion transit time. Likewise, relevant spatial scales can be as small as an electron Debye length or as large as the entire system length. Conventional explicit particle-in-cell (PIC) simulation is restricted by the timestep constraint $\omega_{\mathrm{pe}}\,\Delta t \lesssim 2$ for numerical stability (ω_{pe} is the electron plasma frequency and Δt is the timestep size); hence it is limited to studying microscopic plasma phenomena. More recently, direct implicit (DI) methods were developed [2–5]; these relax the $\omega_{\mathrm{pe}}\,\Delta t$ restraint for stability, allowing particle simulation of large time and space scale physics. With the DI method one still has to control Δt and the grid cell size Δx to accurately resolve the relevant physics. Two important limits on Δt and Δx are

$$\frac{v}{\lambda}\Delta t < \varepsilon_1, \tag{1}$$

$$\frac{\Delta x}{\lambda} < \varepsilon_2, \tag{2}$$

where ε_1 and ε_2 are of order unity. λ is the characteristic electric field scale length ($\lambda^{-1} = \partial_x E/E$), and v is a characteristic velocity of the particles. There are other constraints on accuracy as well (e.g., $\omega\,\Delta t < \varepsilon$) and these will be discussed later in Section 5. Inequality (1) ensures that particles sample the spatial field variation accurately, and Inequality (2) ensures adequate spatial resolution of the grid quantities. These two constraints for resolution of the physical phenomena in time and space couple Δx and Δt.

If the characteristic field scale length λ is small in a local region, but large over the majority of the system length, one must still use the Δt corresponding to the smallest λ when using conventional PIC methods. An example of such an inhomogeneous system is one with a boundary layer (or sheath). In the sheath region, the relevant scale length is the electron Debye length λ_{De}. Using Inequality (1) with the thermal velocity for electrons v_{Te} as a characteristic velocity, we require that $\omega_{\mathrm{pe}}\,\Delta t < \varepsilon_1$. This is a global constraint on Δt for all the particles throughout the entire plasma system, even though the bulk region may be quiescent and have much weaker field gradients. For such cases, the DI method

---

* Present address: Plasma Physics Laboratory, Princeton University, Princeton, New Jersey 08543.

0021-9991/93 $5.00

Copyright © 1993 by Academic Press, Inc.
All rights of reproduction in any form reserved.

Blank

offers no advantage in computing time over the explicit particle methods. In fact the DI method has the disadvantage of being more complex and having a larger computing overhead.

We can take advantage of this localized disparity in time and space scales by moving particles in different regions of phase space with different Δt's. The Δt variation could, for example, be specified using a criterion such as Inequality (1). We term our technique of varying Δt among the particles the *multi-scale* method [1]. This is discussed further in Section 2.

More specifically, this paper deals with applying the multi-scale technique to bounded plasma problems. Bounded systems are naturally suited for the multi-scale method because the sheath that forms at the wall is a short space and time scale structure that may significantly affect the bulk plasma. This bulk behavior would otherwise be dominated by relatively longer space and time scales. One goal is to understand the interaction between the bulk plasma and the sheath.

The model is collisionless, electrostatic, unmagnetized, one dimensional, and bounded, with kinetic ions and electrons. Figure 1 shows a schematic of the bounded multi-scale simulation. The right boundary is a conducting wall that absorbs all particles that come in contact with it. At $x=0$ there is a symmetry plane, where particles are reflected. We specify an initial phase space distribution function: $f(x, v, t=0)$. To test the numerics of both the multi-scale method and the DI boundary conditions we use the following test problem: cutoff Maxwellian electrons at $x=0$, and fixed (or infinitely massive) ions. The system has an analytic solution, so the run may be started from equilibrium. This problem provides a benchmark for the fast time scale electron sheath dynamics. We present results from two bounded problems: the simulation of the sheath problem discussed above, and the simulation of an ion acoustic shock propagating toward an absorbing conducting wall. In the latter case, computing time was reduced by a factor of nine relative to the conventional technique with one fixed Δt.

In Section 2 the multi-scale method is reviewed. In Section 3 our technique for allowing variable grid spacing is outlined. In Section 4 a consistent treatment of the collector sheath boundary using the DI method is presented. In Section 5 we discuss the criterion for specifying particle timesteps. In Section 6 we outline the analytic sheath problem to be modeled. Finally, in Section 7 we show results from the bounded multi-scale simulation for the two bounded test problems with significant savings in computing time.

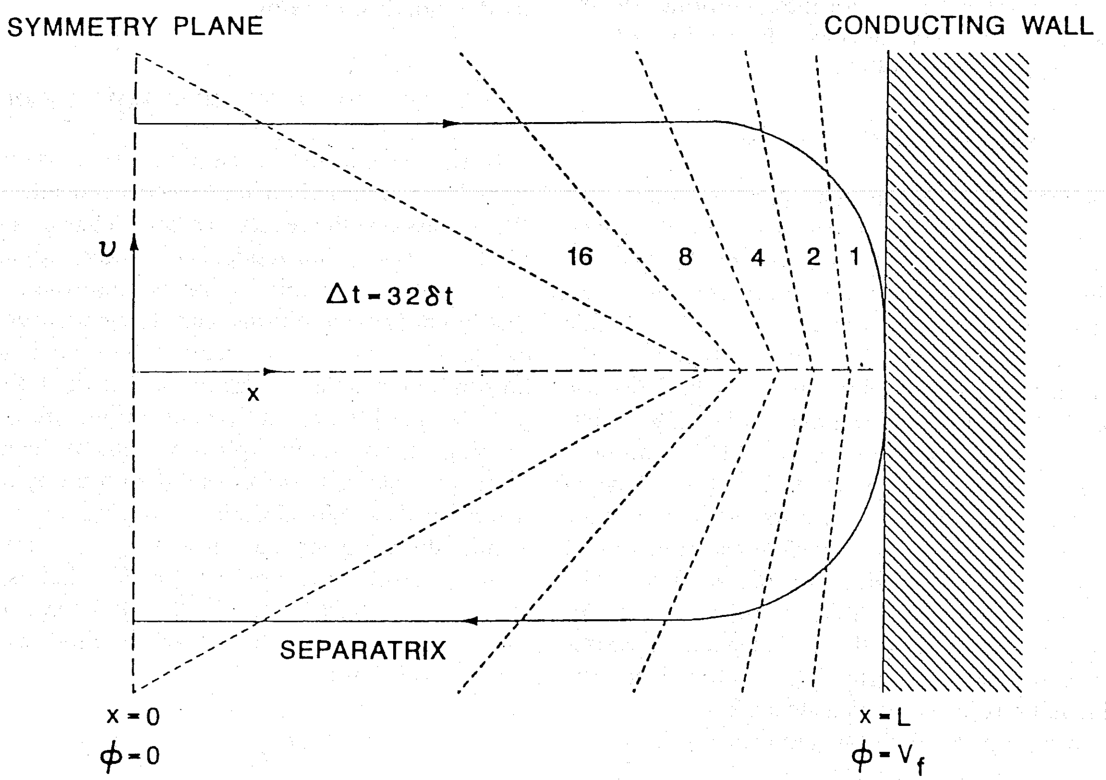

FIG. 1. Schematic of the bounded multi-scale plasma simulation. The case shown has six Δt groups. Particles change Δt as they move in phase space and cross the dotted lines. The solid line represents the separatrix $v_c(x)$, where inside the particles are trapped and outside the particles hit the wall and are absorbed. The ions are fixed and the electrons have a cutoff Maxwellian distribution. The conducting wall is left floating at V_f.

2. THE MULTI-SCALE METHOD

The objective of the multi-scale method is to model macroscopic ($\omega \ll \omega_p$, $\lambda \gg \lambda_D$) plasma phenomena globally, while also accurately treating short time ($\sim \omega_p^{-1}$) and short space scales ($\sim \lambda_D$) in local regions where microscopic physics is important. We accomplish this by allowing groups of particles to move at different Δt's. The multi-scale method is necessarily implicit to allow for $\omega_p \, \Delta t \gg 1$ in the bulk plasma. Each group of particles G_m is pushed every 2^m timesteps using $\Delta t_m = 2^m \, \delta t$, $m = 0, 1, 2, ..., m_{\max}$, where δt is the smallest timestep size. To avoid having occasional timesteps (every 2^m), where we move all the particles in the group G_m at once, we define subgroups of group G_m called blocks B_m^l. Given a group G_m, there are 2^m blocks B_m^l, where $l = 0, 1, 2, ..., (2^m - 1)$. Each block in a group is moved at a different timestep. We move a block B_m^l at timestep n, with $t = n \, \delta t$, when

$$(n) \bmod (2^m) = l. \tag{3}$$

The free streamed charge density $\tilde{\rho}$ and effective susceptibility χ are linearly interpolated at time level n if the block is not moved. Each block has a pair of associated $\tilde{\rho}$ arrays so that interpolation in time is possible. There are $(2^{(m_{\max}+1)} - 1)$ blocks so that for large $m_{\max}$ memory space and computing overhead for these interpolations become items of concern. A more detailed description of the multi-scale method is given in Ref. [1].

3. VARIABLE GRID SPACING

In conventional one-dimensional particle codes, the cpu time for calculation of field quantities is usually trivial compared to the time used for advancing particles. Hence, having a fine grid throughout the entire system is not usually problematic. However, when using the multi-scale algorithm there are $N(2^{(m_{\max}+1)} - 1)$ interpolations of the charge density for each species each timestep, where N is the number of grid cells. Thus, reducing the number of grid cells can greatly reduce the computing overhead. In addition, nonuniform grid spacing permits control of $v \, \Delta t / \Delta x$ when Δt is varied to avoid the associated errors due to numerical heating and cooling [6]. Another benefit is the reduction of noise in regions where fine resolution is not necessary. The noise associated with using a finite number of particles is reduced because increasing the grid spacing increases the number of particles per grid cell, which reduces the statistical fluctuations in the particle density.

To begin setting up the mesh, the number of grid cells N is specified as an input parameter. We also specify the spatial distribution (or density) function $g(x)$, $0 \leqslant x \leqslant L$, which is proportional to the number of grid cells per unit length. Then $g(x)$ is numerically integrated and normalized, producing a cumulative distribution function $G(x)$, with $G(x = 0) = 0$ and $G(x = L) = 1$. Next, a uniform distribution of points, $y_j = j/N$, is inverted to produce the grid point locations, $x_j = G^{-1}(y_j)$. Linear interpolation is used to calculate G^{-1}. This procedure is analogous to loading a given distribution of particles in a conventional PIC code. By using this technique we efficiently produce a variable grid when given the "density" of grid cells needed as a function of x.

Using simplified differencing, the field equation for the DI method is [2–4], $\partial_x (1 + \chi) \, \partial_x \phi = -\tilde{\rho}$, where we have set $\varepsilon_0 = 1$, or equivalently, we have used units where $q \rightarrow q/\varepsilon_0$. Using conservation of electric flux, we arrive at the finite difference equation for the one-dimensional case,

$$\frac{2}{(\Delta x_{j-1/2} + \Delta x_{j+1/2})} \left\{ \frac{(1 + \chi_{j+1/2})}{\Delta x_{j+1/2}} \phi_{j+1} - \left[\frac{(1 + \chi_{j+1/2}}{\Delta x_{j+1/2}} \right. \right.$$
$$\left. \left. + \frac{(1 + \chi_{j-1/2})}{\Delta x_{j-1/2}} \right] \phi_j + \frac{(1 + \chi_{j-1/2})}{\Delta x_{j-1/2}} \phi_{j-1} \right\} = -\tilde{\rho}_j, \tag{4}$$

where we define $\Delta x_{j+1/2} = x_{j+1} - x_j$, $\chi_{j+1/2} = \frac{1}{2}(\chi_{j+1} + \chi_j)$; and $j = 1, 2, 3, ..., J + 1$. Equation (4) is tridiagonal; hence it is easily solved. This equation reduces to the usual finite difference scheme for constant Δx. The boundary conditions used are discussed below.

4. BOUNDARY CONDITIONS FOR THE DI METHOD

In this section we develop consistent boundary conditions for the DI method at an electrostatically floating plate. We will assume the reader has knowledge of the DI method [2–4]. Although our model is one dimensional, we will use vector notation in this section because the formulation of the boundary conditions can be generalized to higher dimensions. Because the "free streaming" charge density $\tilde{\rho}$ is known and not the charge density at the future time level ρ^{n+1}, we need to extend the conventional method [7, 8] for treating boundaries in explicit models. At the wall there is a surface charge σ. This causes a discontinuity in E at $x = L$, making $\partial_x \phi$ ill-defined at that point. Hence, we avoid using a finite difference approximation for $\partial_x \phi$ at $x = L$. This is done by letting the grid be "open"; that is, it becomes infinitesimal close to the wall, but does not include it. Then Gauss' law is used to obtain the following boundary conditions [7, 8]:

$$E(x = L) = \frac{1}{\Delta x} (\phi_J - \phi_{J+1}) + \frac{\Delta x}{2} \rho_{J+1} + \sigma, \tag{5}$$

$$E_{J+1} = \frac{1}{\Delta x} (\phi_J - \phi_{J+1}) + \frac{\Delta x}{2} \rho_{J+1}. \tag{6}$$

For variable grid spacing, $\Delta x = \Delta x_{J+1/2}$. In higher dimensions, one might make the assumption that the tangential component of $\mathbf{E}_{J+1/2}$ is small compared to the perpendicular component and use Eqs. (5) and (6) for the perpendicular component. For the tangential field one can use $\mathbf{E}_t(x=L)=0$.

The above boundary conditions are expressed in terms of ρ_{J+1} and σ at the future time level $n+1$; however, these values are not known. We need Eqs. (5) and (6) in terms of the intermediate values $\tilde{\rho}_{J+1}$ and $\tilde{\sigma}$. The wall charge at the future time level is calculated from

$$\sigma = \tilde{\sigma} + \delta\sigma. \tag{7}$$

Next, we write $\delta\sigma$ in terms of $\delta\mathbf{x}$, which is given in Refs. [2–4] as $\delta\mathbf{x} = \beta(q/m)\,\tilde{\rho}\,\Delta t^2\,\mathbf{E}$,

$$\delta\sigma = \tilde{\rho}_{J+1}\,\delta\mathbf{x}_{J+1}\cdot\hat{\mathbf{n}}, \tag{8}$$

$$= \chi_{J+1}\mathbf{E}_{J+1}\cdot\hat{\mathbf{n}}, \tag{9}$$

where $\chi = \beta(q/m)\,\tilde{\rho}\,\Delta t^2$ and $\hat{\mathbf{n}}$ is the unit vector normal to the surface. Using Gauss' law and the fact that $\mathbf{E}_{J+1} \parallel \hat{\mathbf{n}}$, we obtain $\mathbf{E}_{J+1}$ in terms of σ,

$$\mathbf{E}_{J+1} = -\sigma\hat{\mathbf{n}}; \tag{10}$$

substituting this value into Eq. (9) and inserting the result into Eq. (7), we arrive at

$$\sigma = \frac{\tilde{\sigma}}{(1+\chi_{J+1})}, \tag{11}$$

which can now be used in Eq. (5).

In a similar fashion, we calculate ρ_{J+1} at $n+1$, which is also needed for the boundary conditions Eqs. (5) and (6),

$$\rho_{J+1} = \tilde{\rho}_{J+1} + \delta\rho_{J+1}; \tag{12}$$

$\delta\rho$ is given in Refs. [2–4] as $\delta\rho = -\nabla\cdot(\chi\mathbf{E})$. Evaluating this at $J+1$ we arrive at

$$\delta\rho_{J+1} = -[\nabla_{\perp}\chi]_{J+1}\,E_{J+1} - \chi_{J+1}\rho_{J+1}, \tag{13}$$

where $\nabla_{\perp}$ is $\hat{n}\cdot\nabla$, the gradient perpendicular to the wall. In the one-dimensional case we use a one-sided difference for $\nabla_{\perp}$, because this is the best we can do. This produces ρ at $n+1$ in terms of $\tilde{\rho}$:

$$\rho_{J+1} = \frac{\tilde{\rho}_{J+1}}{1+\chi_{J+1}} + \frac{2(\chi_{J+1}-\chi_{J+1/2})}{\Delta x}\,\frac{\tilde{\sigma}}{(1+\chi_{J+1})^2}. \tag{14}$$

We use $\chi_{J+1/2} = \frac{1}{2}(\chi_J + \chi_{J+1})$ for convenience. In all the simulations presented, particles are demoted to the smallest Δt group when they come close to the wall, making $\chi_{J+1} \ll 1$ and $\rho_{J+1} \approx \tilde{\rho}_{J+1}$. Therefore, in our particular application, the corrections ($\delta\rho$ and $\delta\sigma$) are small.

5. TIME STEP CONTROL

Since particles are allowed to move with different time increments depending on their locations in phase space, we need to develop a criterion for demoting/promoting particles into blocks with larger/smaller timestep sizes. We choose a particle's Δt based on the local truncation error made in the particle advance. The local truncation error gives a measure of how much error is made in the particle advance from one timestep to the next. If this error is too large, we reduce Δt. If this error is smaller than the specified tolerance, then we increase Δt accordingly to increase the computational efficiency.

5.1. *Error Analysis*

For the particle advance, we use the "D1" implicit scheme [2–5]. An analysis of the local truncation error of the D1 scheme is given in the Appendix. In this section we analyze the local truncation error of the leapfrog particle advance because it is analogous to the error calculation of the D1 scheme, but is much more straightforward and also gives the same timestep control criteria. The analysis of the D1 scheme has added complexity because of the recursive filtering in time of the acceleration and the evaluation of the acceleration at an intermediate location (see the Appendix). The leapfrog scheme that we will discuss in this section is [7]

$$v^{n+1/2} = v^{n-1/2} + \Delta t\,a^n(x^n), \tag{15}$$

$$x^{n+1} = x^n + \Delta t\,v^{n+1/2}, \tag{16}$$

where a is the acceleration: $a^n = (q/n)\,E^n(x^n)$. Eliminating v, the scheme can be rewritten as

$$\frac{x^{n+1} - 2x^n + x^{n-1}}{\Delta t^2} = a^n(x^n). \tag{17}$$

We define the quantity x_s^{n+1} as the value of x^{n+1} obtained from the scheme using the exact values $x(t^n)$, $x(t^{n-1})$, and $a(x(t^n), t^n)$ for x^n, x^{n-1}, and a^n. Then we define the local truncation error as the difference between the predicted and actual values: $d^{n+1} \equiv x_s^{n+1} - x(t^{n+1})$ [10]. Using $x_s^{n+1} = x(t^{n+1}) + d^{n+1}$ and substituting the exact quantities into Eq. (17) we obtain

$$\frac{x(t^{n+1}) - 2x(t^n) + x(t^{n-1})}{\Delta t^2} = a(x(t^n); t^n) - \frac{1}{\Delta t^2}\,d^{n+1}. \tag{18}$$

Using Taylor series the left-hand side becomes $\ddot{x}^n(t^n) + (\Delta t^2/12)\, x^{(4)}(t^n) + O(\Delta t^4)$; hence, the local truncation error is

$$d^{n+1} = -\Delta t^4 \tfrac{1}{12}\, \ddot{a}(x^n, t^n) + O(\Delta t^6). \tag{19}$$

In practice we do not use Eq. (17) to advance the particles. Rather than starting with x^{n-1} and x^n and advancing to x^{n+1} as implicitly assumed in Eq. (17), we use $v^{n-1/2}$ and x^n to advance to $v^{n+1/2}$ and $^{n+1}$. For example, one would start the particle motion with $x^0 = x(t=0)$ and some approximate value for $v^{n-1/2}$ as initial conditions. Substituting Eq. (15) into Eq. (16) and the exact values into the scheme except for $v^{n-1/2}$, we obtain

$$x(t^{n+1}) = x(t^n) + \Delta t\, v^{n-1/2} + \Delta t^2\, a(x(t^n), t^n) - d^{n+1}. \tag{20}$$

Depending on what value of $v^{n-1/2}$ is used in Eq. (20), we obtain a different error term. If we use just the right value: $v^{n-1/2} = v(t^n) - (\Delta t/2)\, a(x(t^n), t^n) + (\Delta t^2/6)\, \dot{a}(x(t^n), t^n)$, we maintain the same order error as in Eq. (19). However, if we use $v^{n-1/2} = v(t^n) - (\Delta t/2)\, a(x(t^n), t^n)$, we obtain a lower order error. At $n = 0$, this is the value of $v^{n-1/2}$ that is typically used for starting the particle motion. Using this value for $v^{n-1/2}$ and expanding $x(t^{n+1})$ about t^n we obtain

$$d^{n+1} = -\Delta t^3 \tfrac{1}{6} \dot{a}(x^n, t^n) + O(\Delta t^4). \tag{21}$$

In order to control the error in the particle advance we will keep d^{n+1} small compared to the other two terms in Eq. (20) that cause a change in x. Assuming that the second and third terms on the right-hand side of Eq. (20) do not cancel and that they are not zero, we require

$$\Delta t \left|\frac{\dot{E}}{E}\right| < \varepsilon_1, \tag{22}$$

$$\frac{e}{m} \Delta t^2 \left|\frac{\dot{E}}{v}\right| < \varepsilon_2, \tag{23}$$

for control of accuracy. ε_1 and ε_2 parameterize the amount of error made per timestep. If we assume sinusoidal variation in E then Eq. (22) reduces to the familiar "$kv\, \Delta t$" restriction [3-5] (ω is usually neglected):

$$|kv - \omega|\, \Delta t < \varepsilon_1. \tag{24}$$

Assuming $|\partial_t E| \ll |v \partial_x E|$, Inequality (23) reduces to the "$\omega_{\text{trap}}\, \Delta t$" restriction [3-5]:

$$\frac{e}{m} \Delta t^2\, |\partial_x E| < \varepsilon_2. \tag{25}$$

In our application, it is the particle transit time through the sheath, and not the time-dependent variation of the field, that is the most limiting (the $kv\, \Delta t$ term in Inequality (24)). The $kv\, \Delta t < \varepsilon_1$ criterion is important for our physical applications, but it is difficult to implement adaptively because the statistical fluctuations in E make $k = \partial_x E/E$ a difficult quantity to determine on the grid. For example, if E and $\partial_x E$ are fluctuating about zero (which is typical), there can be spatial locations where $E \approx 0$ and $\partial_x E \neq 0$, producing large values of k. These artificially large values of k are produced by the statistical noise rather than the physics of interest. Inequality (25) could be used for timestep control since $\partial_x E$ could be calculated on the grid, although it may also be a noisy quantity. Smoothing in space and/or time would probably be necessary for most applications.

5.2. *Implementation*

The way particle Δt group demotion/promotion is implemented depends largely on the physics being modeled. For the sheath problem to be presented in Section 7.1, the region that needs fine space-time resolution is stationary so we do not need adaptive timestep control. The technique that we use is to look ahead $Cv\, \Delta t$, where C is a constant and see if the particle will enter a spatial region violating Eq. (25). This produces lines in phase space with slope $C\, \Delta t$ that are the boundaries between Δt groups. We choose $C \geqslant 2$ so that a particle will be promoted up to smaller Δt soon enough to "keep up" with the timestep constraint. That is, $C \geqslant 2$ ensures that a particle will only need to be promoted one group per advance cycle as required by the multi-scale algorithm [1]. Our template for dividing up phase space between Δt groups is kept symmetric about $v = 0$. This demotion/promotion scheme is represented schematically in Fig. 1. The dashed lines represent boundaries between the different Δt groups. The Δt group boundaries (or Δt contours) are shown in Fig. 2C for the actual sheath test runs presented in Section 7.1. The $v\, \Delta t$ look ahead produces a slowly varying χ spatially, which allows a gradual transition from the "implicit region" ($\omega_p\, \Delta t > 1$), to the "explicit region" ($\omega_p\, \Delta t \ll 1$). Although admittedly somewhat ad hoc, this template for breaking up phase space into different Δt groups works quite well.

For the propagating shock front problem to be presented in Section 7.2, we used the same template for the sheath region, but in addition introduced horizontal boundaries between groups, thereby allowing resolution of the shortest wavelengths of interest in the bulk ($kv\, \Delta t < \varepsilon$). Contours of Δt versus location in phase space are shown in Fig. 3 for the electrons in the shock test run. The results of the shock test problem will be discussed in more detail in Section 7.2. This template gave us the greater gains in computing time and better control of accuracy than any of several others [9] which were tested.

When demoting particles (smaller to larger timestep

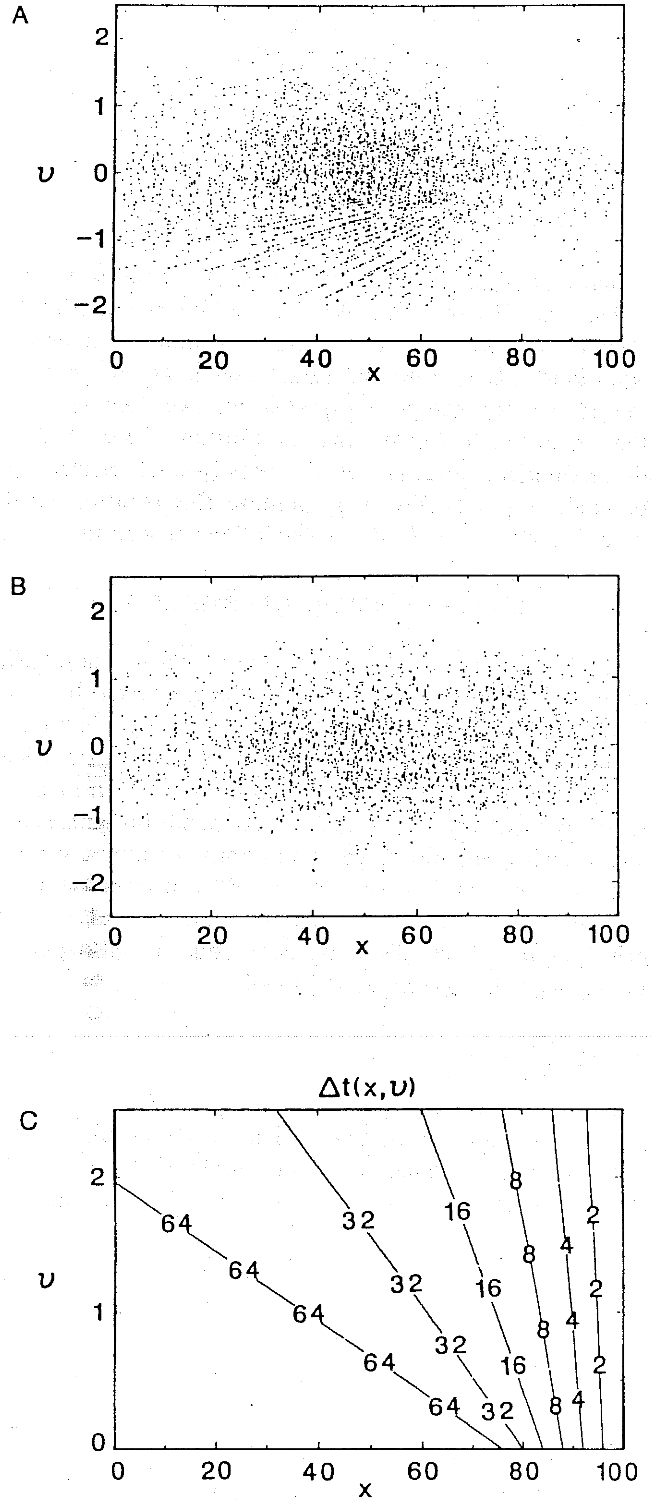

FIG. 2. Snapshot of phase space at $t = 100$ for the sheath test problem. (A) Coherence in phase space caused by immediate demotion when particles cross Δt group boundaries. (B) Delayed demotion showing no coherence. (C) Δt contours in phase space showing boundaries between Δt groups which are fixed in time. Values shown are normalized by the smallest timestep δt.

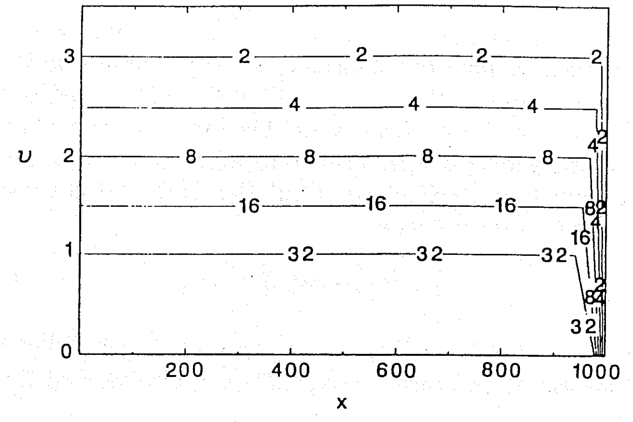

FIG. 3. Electron $\Delta t(x, v)$ group boundaries for the shock test problem. Values shown are normalized by the smallest timestep δt.

sizes), care must be taken to keep from introducing spurious structure into phase space. Assuming no electric field, as a particle crosses a group boundary (demoted from group m to $m+1$), it moves $v\,\Delta t_m$ from the preceding advance; then it moves $2v\,\Delta t_m$. This causes a "wedge" of particles in phase space which previously crossed the group boundary with displacement $v\,\Delta t_m$ to move $2v\,\Delta t_m$ on the next advance, leaving an empty wedge $v\,\Delta t_m$ in width. The affect on phase space is shown in Fig. 2A for the sheath test problem discussed in Section 7.1. This phase space structure did not significantly affect our simulation results presented in Section 7 (except for the appearance of the phase space scatter plots). The error is alleviated by delaying the promotion of half of the particles (those with odd array index) until the next timestep. This delay fills out the empty wedge in phase space and removes the coherence, as shown in Fig. 2B.

6. PHYSICS OF THE SHEATH PROBLEM

In this section we develop an analytically solvable equilibrium sheath model which is used to benchmark the bounded multi-scale code. We assume fixed background ions and model only the fast time scale electron motion at the sheath. There is no plasma flow to the plate; hence, no plasma source is necessary for an equilibrium. The distribution function for the electrons at the center of the system $f(x=0, v)$ is specified as a cutoff Maxwellian symmetric about $v = 0$,

$$f(x, v) = \begin{cases} \dfrac{n_0}{2\sqrt{2}\, v_T \operatorname{erf}((\sqrt{2}/2)(v_0/v_T))} \\ \qquad \times \exp\left\{-\dfrac{1}{2}\dfrac{v^2}{v_T^2} + \dfrac{e\phi(x)}{T}\right\}; & |v| < v_c(x), \\ 0; \qquad |v| \geqslant v_c(x), \end{cases} \tag{26}$$

where $v_0 = v_c(x=0)$, $n_0 = n_e(x=0)$, and $mv_T^2 = T$ are specified as input parameters; $v_c(x)$ is the cutoff velocity of the electrons with $v_c(x) \geqslant 0$. Equation (26) satisfies the Vlasov equation at steady state. The ions are assumed to be uniform and fixed: $n_i(x) = n_0$. With $\partial/\partial t = 0$, we have energy conservation which allows calculation of the cutoff velocity in terms of the potential:

$$\tfrac{1}{2}mv_c^2(x) - e\phi(x) = \tfrac{1}{2}mv_0^2, \tag{27}$$

We can now write Poisson's equation with the following dimensionless variables $u = (\sqrt{2}/2)(v/v_T)$ and $\psi = e\phi/T$ as

$$\frac{d^2\psi}{dx^2} = \frac{1}{\lambda_D^2}\left\{\frac{e^{\psi}\,\mathrm{erf}\sqrt{u_0^2+\psi}}{\mathrm{erf}(u_0)} - 1\right\}. \tag{28}$$

We can solve this ordinary differential equation numerically as an intial value problem starting at $x = L$ and "advancing" backwards to $x = 0$. To do this when we need to know the two boundary conditions: $\psi(x=L)$ and $(d\psi/dx)|_{x=L}$. We start at $x = L$ because of difficulties with convergence starting at $x = 0$. The first boundary condition is obtained using Eq. (27) with $v_c(x=L) = 0$,

$$\psi(x=L) = -u_0^2, \tag{29}$$

where $u_0 = u(v_0)$. Next, we multiply Eq. (28) by $d\psi/dx$, and use

$$\frac{1}{2}\frac{d}{dx}\left(\frac{d\psi}{dx}\right)^2 = \frac{d^2\psi}{dx^2}\frac{d\psi}{dx};$$

then integrate over the system length to obtain the second boundary condition,

$$\left.\frac{d\psi}{dx}\right|_{x=L} = \frac{\sqrt{2}}{\lambda_D}\left\{\int_0^{\psi(x=L)}\left[\frac{e^{\psi'}\,\mathrm{erf}\sqrt{u_0^2+\psi'}}{\mathrm{erf}(u_0)} - 1\right]d\psi'\right\}^{1/2}. \tag{30}$$

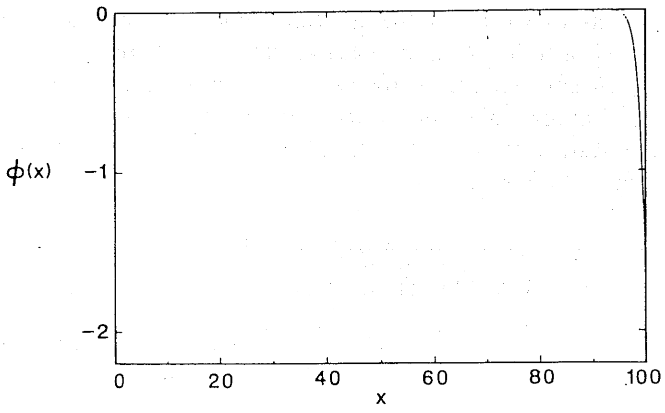

FIG. 4. Numerical solution of the electrostatic potential, which gives the initial conditions for the sheath test problem.

TABLE I

Parameters for Sheath Test Run

$q/m = -1$	$\omega_p = 1$	$\lambda_D = 1$	$v_T = 1$	$v_c = 2$
$L = 100$	$N_0 = 40{,}000$	$\Delta x = 0.25 - 4$	$\delta t = 0.1$	$G = 1, 64$

Equation (28) can now be solved using the boundary conditions, Eqs. (29) and (30). We use the Nyström method for numerically solving second-order ordinary differential equations [10]. Knowing $\phi(x)$ we load the particles (electrons) according to Eq. (26) and we then can start the simulation off from this equilibrium. Figure 4 shows the numerical solution of the steady-state electrostatic potential Eq. (28). We will compare this solution to the time-dependent simulation in the following section.

7. RESULTS FROM THE BMS CODE

A bounded multiscale particle-in-cell plasma simulation code, BMS, was developed to test ideas presented here and in Ref. [1]. "Proof of principle" for the multi-scale method is shown by making self-consistent runs allowing particles to change Δt groups depending on their location in phase space. We use timestep control that depends on time, space, and velocity. Significant gains in computer time are made over conventional PIC simulations. BMS has the following features: DI method, bounded, multi-scale, and variable grid spacing. The BMS particle code is electrostatic, unmagnetized, and one dimensional.

7.1. *Sheath Test*

We first test the multi-scale method using the sheath problem discussed in Section 6. The system is started from equilibrium with a constant background ion density and the electron distribution specified by Eq. (26). No source is

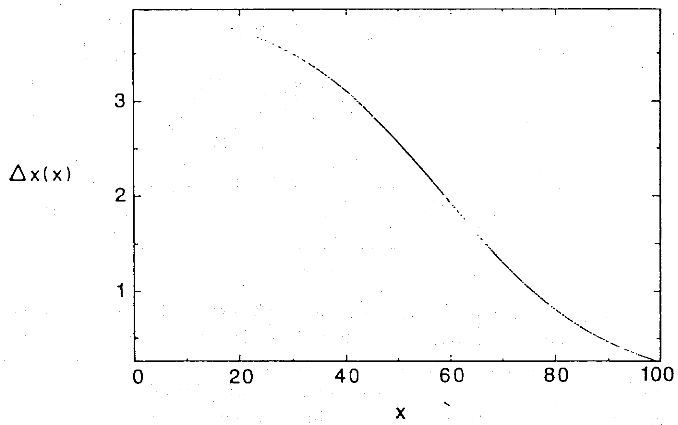

FIG. 5. Variable grid spacing: Δx vs x.

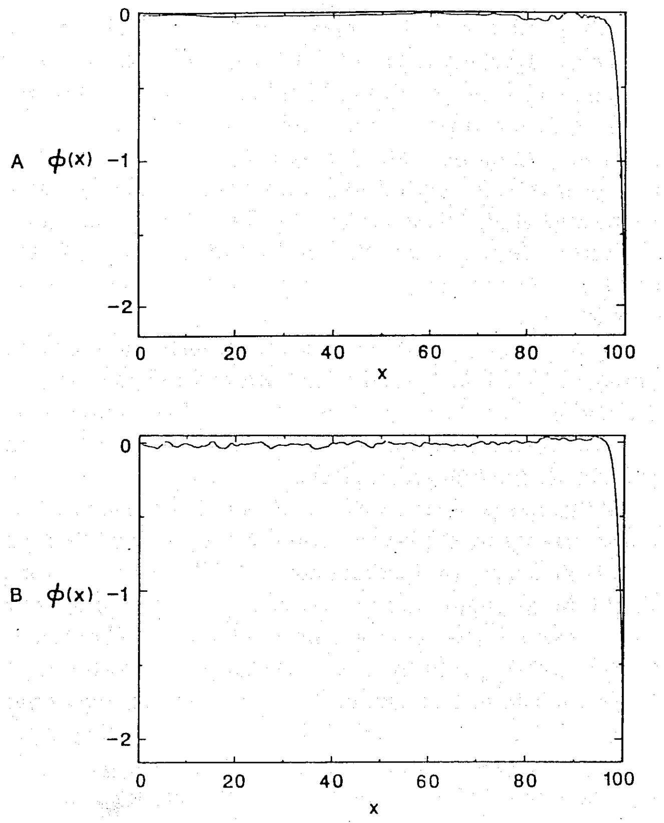

FIG. 6. Snapshots of electrostatic potential versus x at $t = 100$: (A) Multigroup $G = 64$ case; (B) One-group $G = 1$ case.

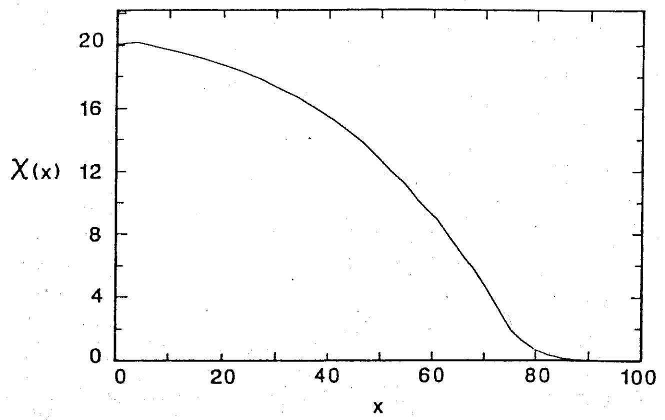

FIG. 7. Snapshot of χ versus x at $t = 100$ for the multigroup $G = 64$ case. For uniform density (which is the case in the bulk region), χ gives an estimate of how "implicit" the region is.

present. For comparison, we made two runs, one with one Δt group ($\omega_p \Delta t = 0.1$) and one with multiple Δt groups ($\omega_p \Delta t = 0.1$–6.4). The test run parameters are given in Table I. We use the MKSA units system, but with ε_0 set to unity. Alternately, the units can be described as dimensionless with length and time normalized by λ_D and ω_p^{-1}. We define the gain G as the ratio of largest to smallest timestep size ($G \equiv \text{largest } \Delta t / \delta t$), where δt is the smallest timestep. Figure 2B shows a snapshot of the phase space at $t = 100$. Figure 2C shows the Δt contours as a function of (x, v) (or boundaries between the different Δt groups). These contours are fixed in time. Figure 5 shows the variable grid spacing, Δx versus x as discussed in Section 3. Figure 6 are snapshots of the electrostatic potential for: A the multigroup $G = 64$ case and B the one-group $G = 1$ case. Note that the $G = 1$ (small $\omega_{pe} \Delta t$) run has more short wavelength fluctuations in the bulk since plasma oscillations are resolved globally. Figure 7 is a snapshot of χ for the $G = 64$ case. The magnitude of χ gives an idea of how implicit the spatial region is in the system ($\chi = \frac{1}{2}\omega_p^2 \Delta t^2$). Figure 8 is the time history of the total potential drop across the system ($\phi(x = L, t)$) for A the multigroup $G = 64$ case and B the one-group $G = 1$ case. The total energy error was small in both cases (less than 1%). Figures 8A and B show that the wall potential oscillates at the plasma frequency about the correct equilibrium value, which is two. We suggest some causes of these noticeable oscillations. First, the simulation was not started exactly at equilibrium because of errors associated with loading the distribution Equation (26) with a finite number of particles and grid

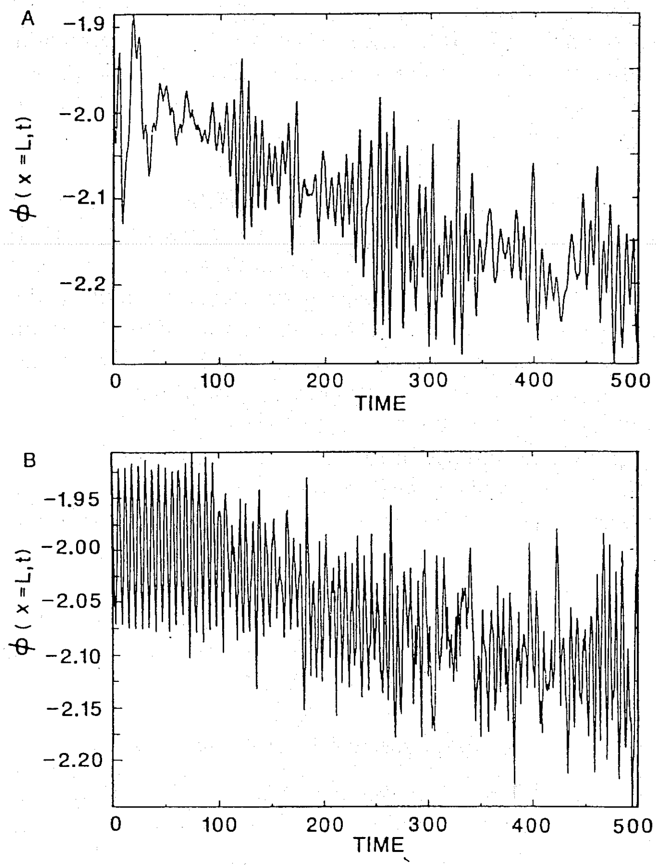

FIG. 8. Time history of the total potential drop for the sheath runs: (A) Multigroup $G = 64$ case; (B) One-group $G = 1$ case.

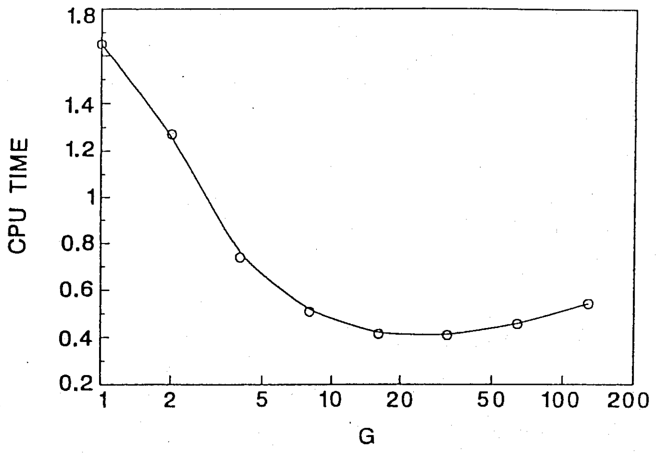

FIG. 9. CPU time versus G for the sheath test problem. G ranges from 1 to 128; the optimum value (minimum cpu time) is at $G = 32$.

cells. Second, the many particle simulation has time dependent fluctuations not taken into account in the equilibrium calculation. Both these differences between the equilibrium theory and the simulation are accentuated by the fact that in the sheath region there is a very low particle density and hence it is more discrete. That is, the simulation exhibits many-particle behavior rather than continuum Vlasov behavior. It is also observed that the potential gradually drops as a few particles leave the system (about one every eight ω_p^{-1}). The many particle simulation is slightly collisional [7], causing particles to diffuse across the separatrix and hit the wall (see Fig. 1).

Figure 9 shows the cpu time per particle per smallest timestep in microseconds versus G for eight runs, with G ranging from one to 128. All calculations of computing time were done on a Cray-2 supercomputer. In calculating the cpu time we subtracted the time associated with setting up the initial conditions and performing diagnostics. Note that there is an optimum value (minimum cpu time) at $G = 32$. Beyond this point, the nonvectorized sorting of particles, along with the interpolation of $\tilde{\rho}$ in time for each block, begins to dominate the computer time. Increasing the number of particles will reduce this effect by increasing the fraction of cpu time used in the particle advance. Also, decreasing the number of grid cells will enhance the gain since $(2^{(m_{\max}+1)} - 1)N$ interpolations in time of $\tilde{\rho}$ are made each timestep, where N is the number of grid cells. The $G = 32$ run was four times faster than the $G = 1$ run.

7.2. *Shock Front Propagating toward a Conducting Wall*

As a second test of the bounded multi-scale formulation, we simulate the propagation of an ion acoustic shock front toward a conducting absorbing plate. This problem is inspired by problems involving transient plasma flow in the divertor region of a tokamak fusion reactor. The edge layer of a diverted tokamak exhibit bursts of MHD activity, known as edge localized modes (ELMs [11] which produce (and are characterized by) sudden bursts of plasma flowing to the divertor region. Our model can be used to study transient plasma flow along magnetic field lines. We start with an intial slab of hot dense plasma and a colder low density background. The details of this flow may be important to divertor design, since the ion energies directed into the divertor plate may be greater during such a transient than at steady state.

The ion acoustic shock has been previously analyzed and simulated [12, 13]. Mason's simulations [13] used kinetic ions and isothermal Boltzmann electrons. In our model we use fully kinetic electrons with differing temperatures and also include the boundary effects.

For this test problem, we load as initial conditions a low density background plasma (component A), and then we add a high density slab (component B). All distributions are initially Maxwellian, and no source is present. The density ratio between the two components is 10 (n_B/n_A). The system length is $1000\lambda_{De}$ with the dense slab covering two-thirds of the system length. The temperature ratios are the following: $T_e^B = 100T_e^A$, $T_i^A = T_e^A$, and $T_i^B = T_i^A$, where the subscripts (i, e) specify the species, and the superscripts (A, B) specify the component. The parameters for the run are listed in Table II. We use the same units as in Section 7.1 (i.e., time in units of ω_{pe}^{1} and length in units of λ_{De}, in terms of the hot electron component). This choice of parameters provides a simulation that is rich with kinetic physics.

We define the thermal transit time from the slab edge to the wall of the hot dense electrons as: $\tau_e^B \equiv (L - L_{\text{slab}})/v_{Te}^B = 400$. Figure 10 shows snapshots of the ion charge density at: A, $t = 100$, the initial expansion; B, $t = 400$, the dense electron thermal transit time $t = \tau_e^B$; and C, $t = 2000$, the time just prior to when the front hits the wall. The solid line with the value of 1.1 at $x = 0$ is the total ion density, the dashed line with the value of 1.0 at $x = 0$ is

TABLE II

Parameters for Sheath Test Run

Parameters	$\Delta x = 0.25 - 2$ $L = 1000$	$\delta t = 0.2$ $L_{\text{slab}} = 600$	$T = 2000$
Background electrons A	$\omega_p = 0.316$ $N = 100{,}000$	$q/m = -1$ $G = 32$	$v_T = 0.1$
Background ions A	$\omega_p = 0.0316$ $N = 100{,}000$	$q/m = 0.01$ $G = 128$	$v_T = 0.01$
Slab electrons B	$\omega_p = 1$ $N = 300{,}000$	$q/m = -1$ $G = 32$	$v_T = 1$
Slab ions B	$\omega_p = 0.1$ $N = 300{,}000$	$q/m = 0.01$ $G = 128$	$v_T = 0.01$

the dense slab ion density (component B), and the dashed line with the value of 0.1 at $x = 0$ is the background plasma ion density (component A). At later times ($t > 2000$), the front hits the wall and is simply absorbed. Figure 11 shows snapshots of the spatial profile of the electrostatic potential at: A, $t = 100$; B, $t = 400$; and C, $t = 2000$. Figure 12 is the time history of the total potential drop across the system. For $t = 100$ the time elapsed is less than τ_e^B. At this point in time the electrons have not yet reached the wall and the collector sheath has not yet formed (Fig. 11A)). At $t = 400$, the time elapsed is τ_e^B and the collector sheath has developed (Fig. 11B)).

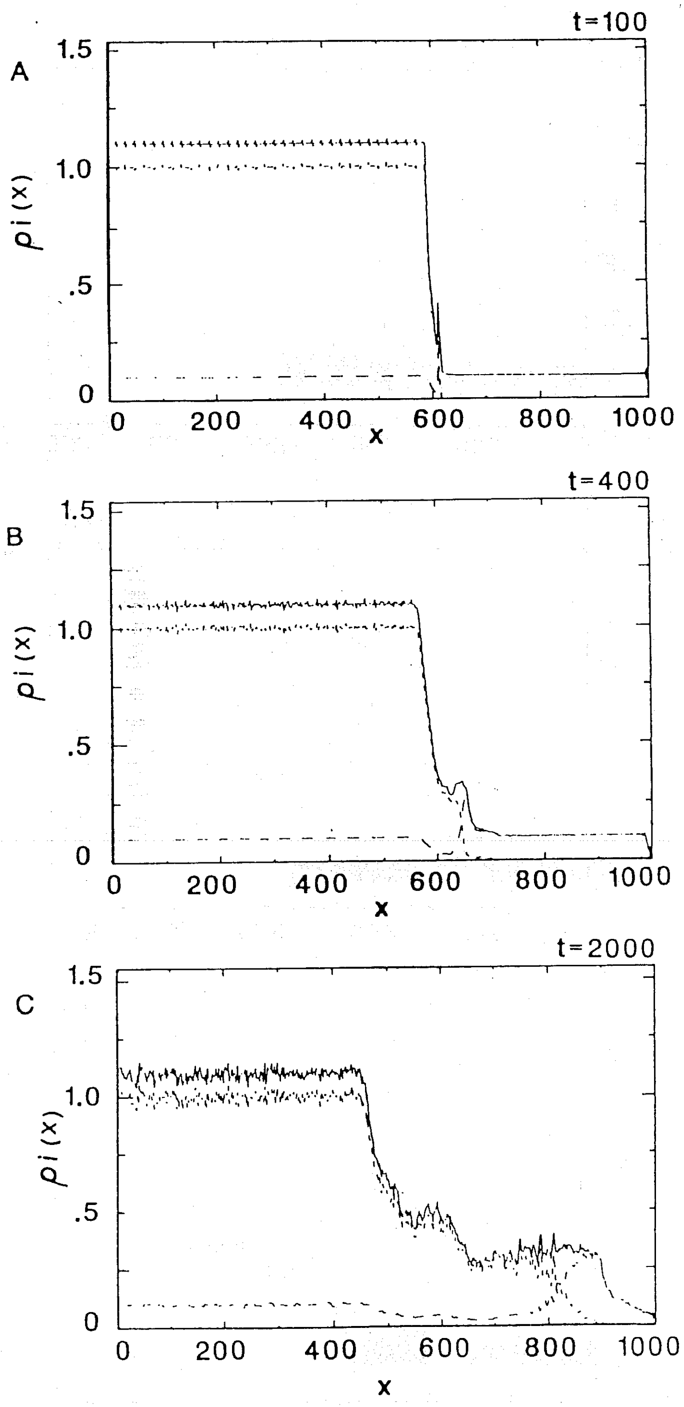

FIG. 10. Snapshots of the charge density for the ions at: (A) $t = 100$; (B) $t = 400$; (C) $t = 2000$. The solid line is the total density, and the dashed lines are the two components, the dense ions and the background ions.

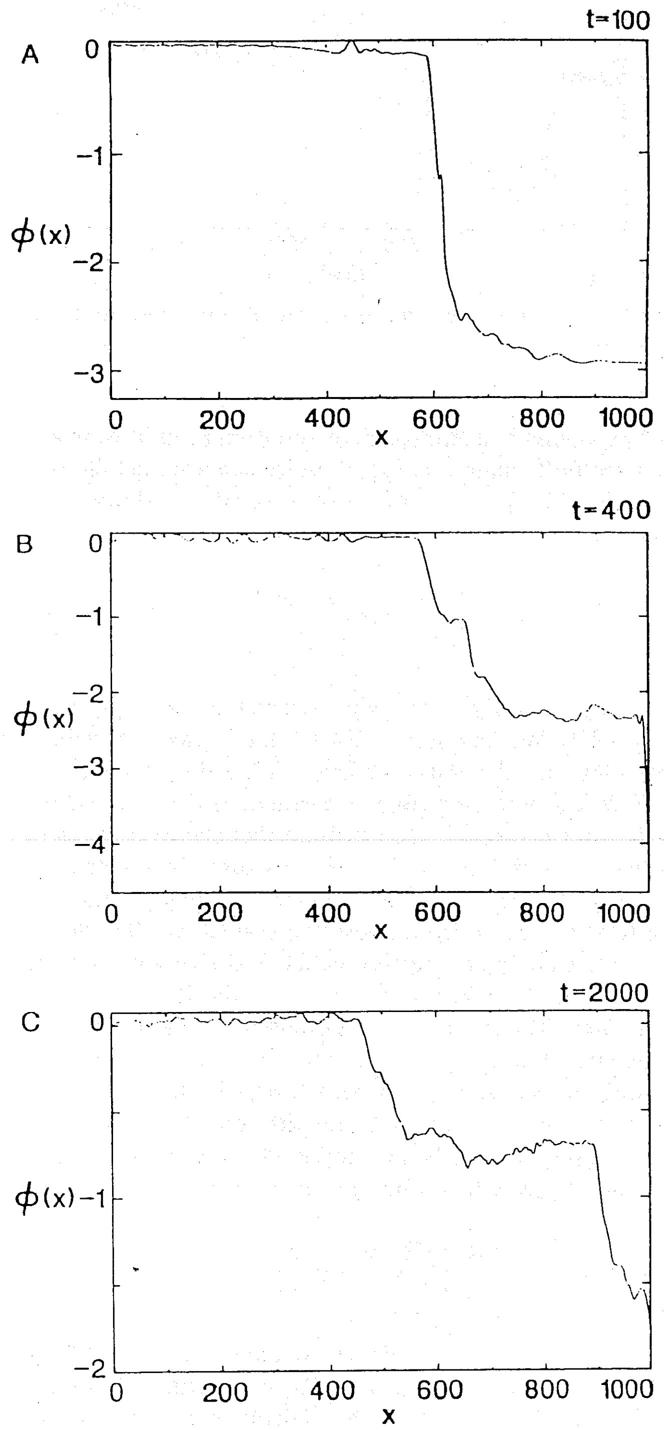

FIG. 11. Snapshots of the electrostatic potential for the shock problem at: (A) $t = 100$; (B) $t = 400$; (C) $t = 2000$.

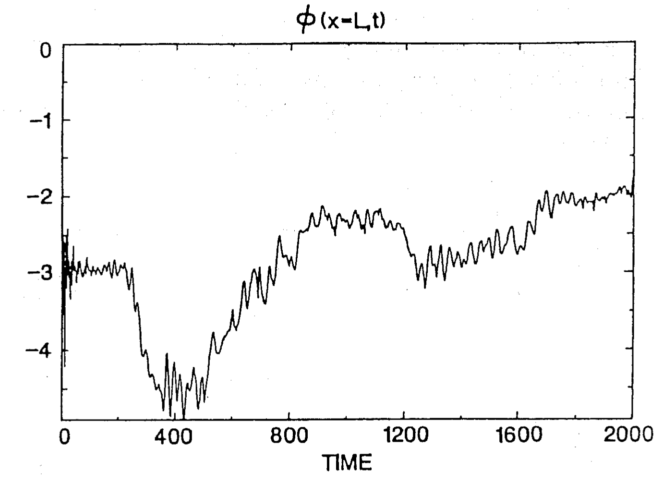

FIG. 12. Time history of the total potential drop versus time for the shock problem.

If we assume the electrons are in thermal equilibrium and obey the Boltzmann relation, then we can use the following expression for the potential drop across the shock front $\Delta\phi$ [13]:

$$\frac{e\,\Delta\phi}{T_e} = -\ln\left(\frac{n_A + n_B}{n_A}\right). \tag{31}$$

Assuming $T_e = T_e^B$, the value predicted by Eq. (31) is $\Delta\phi = -2.3$. We compare this to the measured values: $\Delta\phi(t=100) \approx -2.5$, $\Delta\phi(t=400) \approx -2.3$, and $\Delta\phi(t=2000) \approx -1.4$. With increasing time, the potential drop across the shock front decreases. We attribute this decrease of $\Delta\phi$ to the cooling of the electrons, which is caused by preferential absorption at the wall of the electrons with highest velocity. The total potential drop across the system can be crudely estimated using an expression which is commonly used for determining the collector sheath potential drop [14, 15]. First, we assume that the potential is monotonically decreasing. We also assume that the flux of the ions entering the shock front (where $\phi = 0$) is the result of a half-Maxwellian distribution and that the electrons are in thermal equilibrium. Then we equate the flux of the ions and electrons at the wall and obtain the following result:

$$\frac{e\phi(x=L)}{T_e} = -\ln\left(\frac{v_{Te}}{v_{Ti}}\right). \tag{32}$$

Assuming $T_e = T_e^B$, the value predicted by Eq. (32) is $\phi(x=L) = -4.6$, which compares well with the values around $t \approx \tau_e^B$. For $t > \tau_e^B$, the total drop decreases, possibly due to the decrease in the temperature of the electrons.

Figure 13 shows the dense (component B) ion phase space at: A, $t = 100$; B, $t = 400$; and C, $t = 2000$. The sound speed is $c_s \equiv \sqrt{T_e^B/m_i} = 0.1$ for this run. It is seen from Fig. 13 that most of the shock region ions are accelerated to velocities $v \approx 1.1$–1.5. This agrees reasonably well with the analysis of Moiseev and Sagdeev neglecting the density of reflected ions, which predicts that the shock front is moving at a speed greater than c_s, but less than $1.6c_s$ [16, 17].

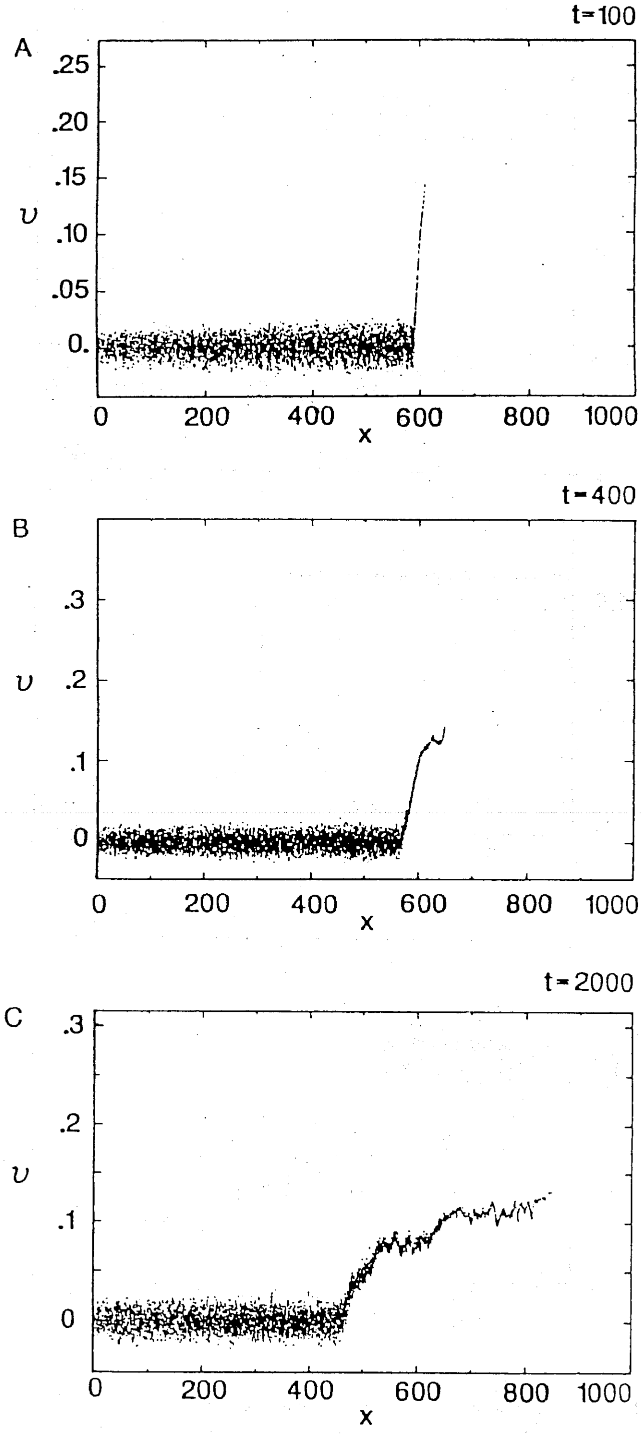

FIG. 13. Snapshots of phase space for the dense slab of ions at: (A) $t = 100$; (B) $t = 400$; (C) $t = 2000$.

Figure 14 shows the low density background (component A) ion phase space at: A, $t = 100$; B, $t = 400$; and C, $t = 2000$. Figures 13 and 14 show that there is a low density leading edge of high energy ions with particles accelerated to velocities as high as $3c_s$. For $t > 2000$ the high energy ions hit the wall and are absorbed. No further high energy ions are observed.

Figure 15 shows the hot dense (component B) electron phase space at: A, $t = 100$; B, $t = 400$; and C, $t = 2000$ (any apparent horizontal lines in the phase space are due to printing errors). At $t = 100$, the electrons have not yet reached the wall. At $t = 400$, the time elapsed is the thermal transit time τ_e^B. At this time, the electrons have reached the wall and are reflected by the sheath potential drop.

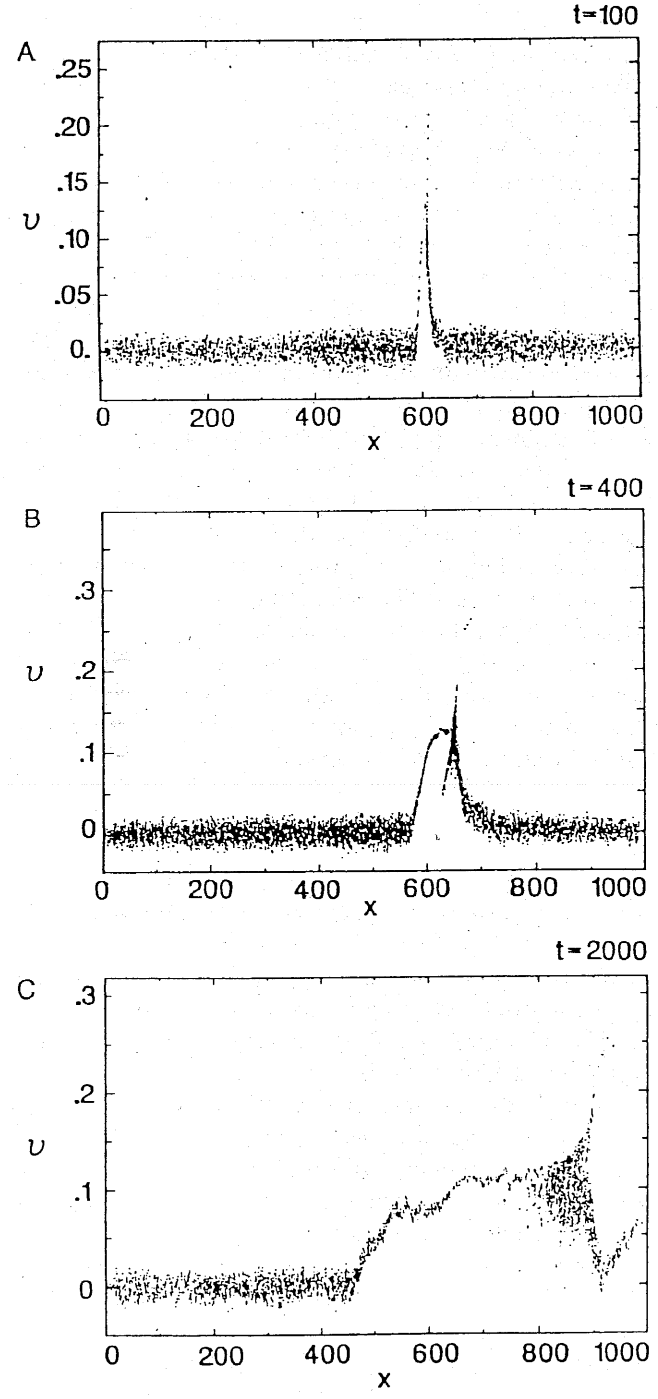

FIG. 14. Snapshots of phase space for the background ions at: (A) $t = 100$; (B) $t = 400$; (C) $t = 2000$.

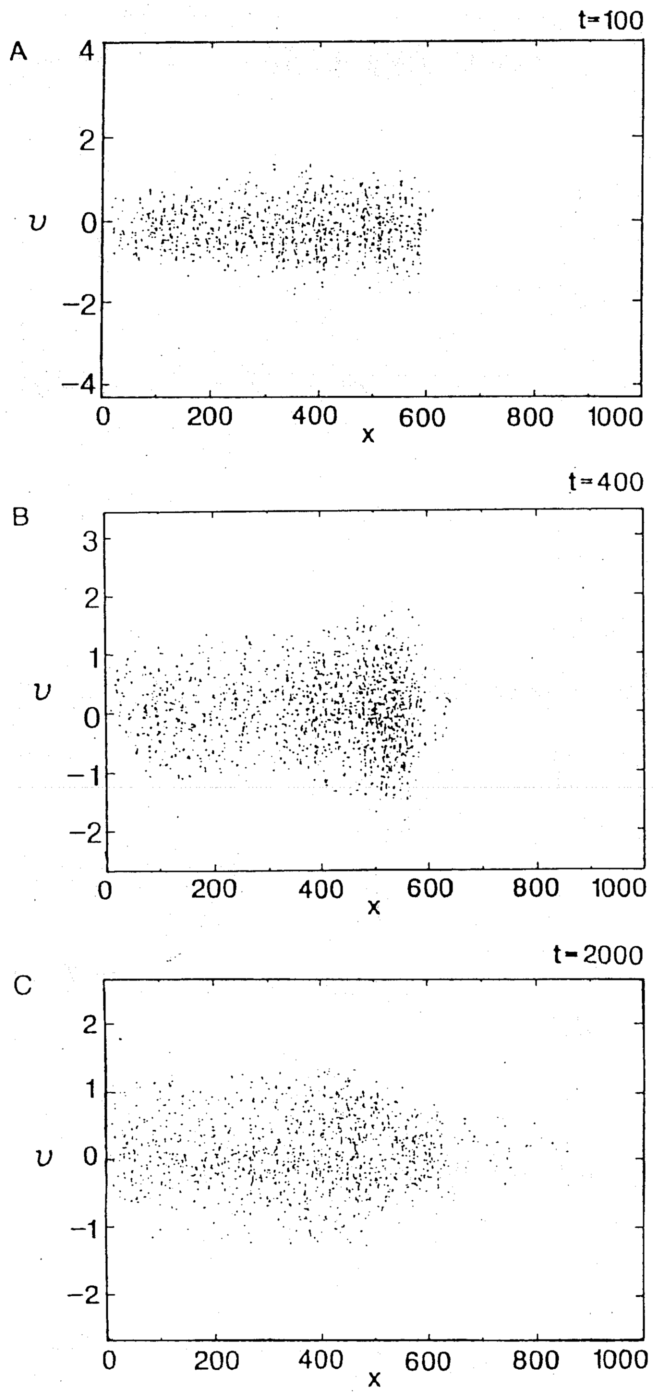

FIG. 15. Snapshots of phase space for the hot, dense slab of electrons at: (A) $t = 100$; (B) $t = 400$; (C) $t = 2000$.

Figure 16 shows the low density background (component A) electron phase space at: A, $t=100$; B, $t=400$; and C, $t=2000$. At $t=100$ the electrons in the shock region are accelerated backwards to a velocity of about -2 (which is $-2v^{B}_{Te}$). At $t=400$, more electrons are accelerated backward through the shock front potential drop and a convective beam-plasma instability occurs, leading to the formation of phase space vortices as the instability saturates. The electrons with $v \approx +2$ are from the reflection of left-going particles as they pass through the symmetry plane at $x=0$. At $t=2000$, the electron phase space begins to "fill in" (or thermalize) due to turbulence, but there is still a noticeable empty region.

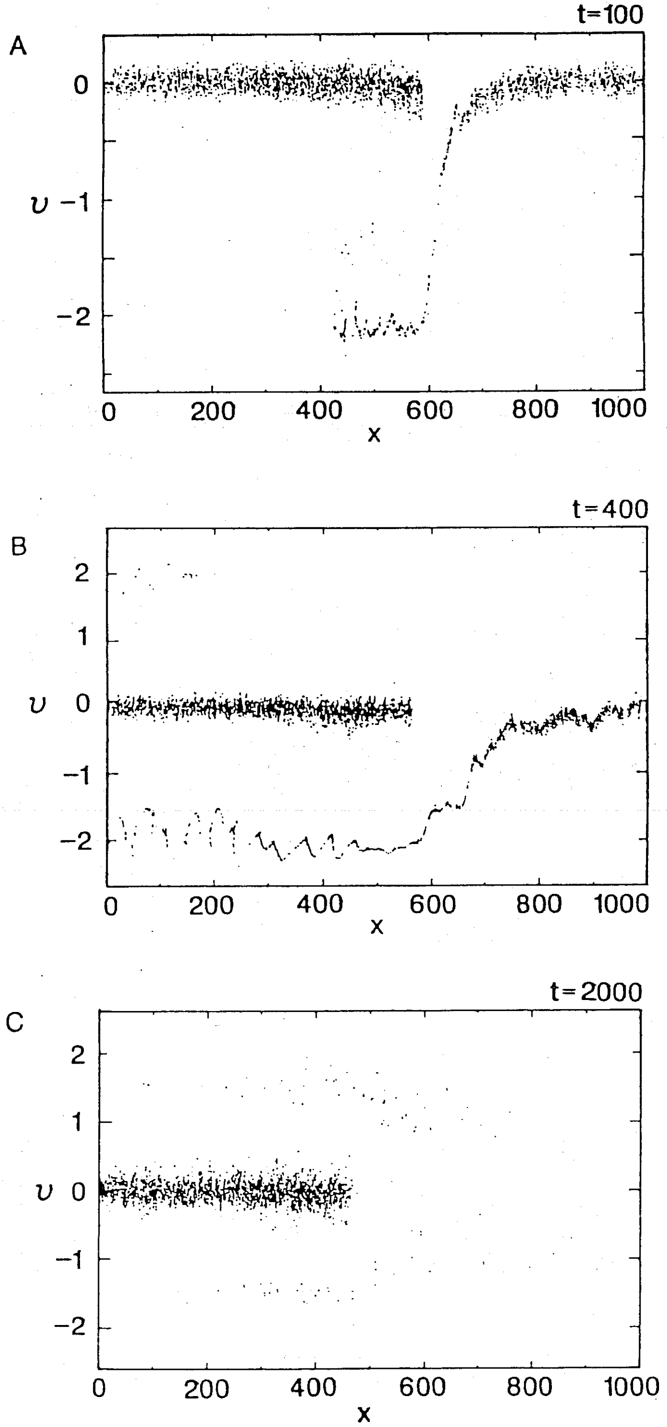

FIG. 16. Snapshots of phase space for the background electrons at: (A) $t=100$; (B) $t=400$; (C) $t=2000$.

In all the phase space snapshots discussed above, 20,000 particles were plotted per snapshot. Hence, the density of dots does not give a comparison of density between the background A component and the dense slab B component (which are different scatter plots).

As seen in Fig. 12, there is a sudden drop in the wall potential at the beginning of the simulation. Because of this initial transient, we do not begin to demote (to larger Δt) particles at $t=0$, but wait a period of time ($t=40$) and then gradually demote particles to larger timesteps. This provides better temporal resolution of the initial transient. The timing for this run was 0.18 μs *per particle per smallest timestep*, including all computation except initial conditions and diagnostics. There was a *gain of nine* over the one-group run (which has a cpu time of 1.65 μs per particle per timestep). The simulation results we present here were benchmarked against one-group small $\omega_{pe}\,\Delta t(=0.2)$ run to the time $t=800$ (4000 timesteps). We did not run the one-group case as long as the multigroup case because of the computing expense. The one-group run used a total (including diagnostics and initial conditions) of 2.56 cpu hours. The multigroup run was run to $t=4000$ (20,000 timesteps) and used total 1.47 cpu hours. There was a -3% change in the total energy for the multigroup run due to damping from using the DT scheme.

8. DISCUSSION

We have been successful in applying the multi-scale method to a sheath problem that is representative of the typical short time scale physics at a boundary. We have also simulated an ion acoustic shock front propagating towards a conducting wall with a significant reduction in computing time. The measured cpu time was 0.18 μs per particle per timestep on a Cray-2 supercomputer, which was a gain of nine over the one-group run. We have implemented a variable Δx grid and developed consistent boundary conditions for bounded DI simulations. Also developed were timestep control criteria, and our prescription for demoting/promoting particles between Δt groups. These criteria were successfully tested in our two examples with demotion/promotion depending on time, position, and velocity.

In higher dimensional electrostatic problems with more complex geometry and boundaries, one could use an unstructured mesh [18] and control the timestep based on how quickly a particle crosses the local grid cell. However, for the magnetized case the gyrofrequency would still need to be resolved [19], unless a reduced physics model is used (e.g., drift-kinetics or gyrokinetics). In the future, it may be possible to extend the ideas presented in this paper to electromagnetic implicit simulations [20].

APPENDIX: LOCAL TRUNCATION ERROR OF THE D1 SCHEME

In this appendix we analyze the local truncation error of the D1 scheme. This analysis is analogous to that of the leapfrog scheme given in Section 5.1. There is added complexity in this calculation due to the recursive filtering in time of the acceleration Eq. (34), and the use of the free streaming position $\tilde{x}$ for calculation of the acceleration. Analysis of the stability and accuracy of the D1 scheme in terms of plasma dispersion is given in Ref. [5]. The D1 scheme centered at integer time levels [1] is the following:

$$\tilde{x}^{n+1} = x^n + \Delta t\, v^n, \tag{33}$$

$$\bar{a}^n = \tfrac{1}{2}(\bar{a}^{n-1} + a^{n+1}), \tag{34}$$

$$v^{n+1} = v^n + \frac{\Delta t}{2}(\bar{a}^n + a^{n+1}), \tag{35}$$

$$x^{n+1} = \tilde{x}^{n+1} + \frac{\Delta t^2}{2} a^{n+1}. \tag{36}$$

For the DI method we use $a^{n+1} = a^{n+1}(\tilde{x}^{n+1})$, not $a^{n+1} = a^{n+1}(x^{n+1})$, because we do not know this quantity in advance of when we use it in Eq. (36). Eliminating v and $\tilde{x}$ partially, the scheme can be rewritten as

$$\frac{x^{n+1} - 2x^n + x^{n-1}}{\Delta t^2} = \frac{1}{4} \sum_{i=-1}^{n} \frac{a^{n-i}(\tilde{x}^{n-i})}{2^i}. \tag{37}$$

In order to completely eliminate $\tilde{x}$, we use Eq. (36) and expand about x^{n-i}:

$$\begin{aligned} a^{n-i}(\tilde{x}^{n-i}) &= a^{n-i}\left(x^{n-i} - \frac{\Delta t^2}{2} a^{n-i}(\tilde{x}^{n-i})\right), \\ &= \left\{1 - \frac{\Delta t^2}{2} \partial_x a^{n-i}(x^{n-i})\right\} a^{n-i}(x^{n-i}) + O(\Delta t^4). \end{aligned} \tag{38}$$

Substituting Eq. (38) into Eq. (37), we can write the scheme as

$$\begin{aligned} \frac{x^{n+1} - 2x^n + x^{n-1}}{\Delta t^2} = \frac{1}{4} \sum_{i=-1}^{n} \frac{1}{2^i} \left\{1 - \frac{\Delta t^2}{2} \partial_x a^{n-i}(x^{n-i})\right\} \\ \times a^{n-i}(x^{n-i}) + O(\Delta t^4). \end{aligned} \tag{39}$$

We define the quantity x_s^{n+1} as x^{n+1} obtained by using the scheme Eq. (39) with the exact values used for x^n, x^{n-1}, and $a(x^{n-i})$, $i = -1, 0, 1, 2, ..., n$. Then we define the local truncation error as $d^{n+1} \equiv x_s^{n+1} - x(t^{n+1})$ [10]. The error d^{n+1} is calculated by substituting $x(t^{n-i})$ for x^{n-i} and $a(t^{n-i}, x(t^{n-i}))$ for a^{n-i} into Eq. (39), and expanding about t^n:

$$\begin{aligned} &\frac{x(t^{n+1}) - 2x(t^n) + x(t^{n-1})}{\Delta t^2} \\ &= a(x(t^n), t^n) + t^2 \ddot{a}(x^n, t^n) \\ &\quad - \frac{\Delta t^2}{2} a(t^n, x(t^n))\, \partial_x a(t^n, x(t^n)) \\ &\quad + O(\Delta t^3) - \frac{1}{\Delta t^2} d^{n+1}, \end{aligned} \tag{40}$$

where we have assumed n large in calculating the second and third terms on the right-hand side. Using Taylor series the left-hand side becomes $\ddot{x}^n(t^n) + (\Delta t^2/12)\, x^{(4)}(t^n) + O(\Delta t^4)$; hence, the local truncation error is

$$\begin{aligned} d^{n+1} = \Delta t^4 \left\{\tfrac{11}{12}\ddot{a}(x^n, t^n) - \tfrac{1}{2} a(t^n, x(t^n))\, \partial_x a(t^n, x(t^n))\right\} \\ + O(\Delta t^5). \end{aligned} \tag{41}$$

We do not use Eq. (37) to advance the particles. Rather than starting with x^{n-1} and x^n and advancing to x^{n+1} as delineated in Eq. (37), we use x^n and v^n to advance to x^{n+1} and v^{n+1}. For example, one would start the particle motion with $x^0 = x(t=0)$ and $v^0 = v(t=0)$ as initial conditions. Substituting Eq. (33) into Eq. (36), and $x(t^n)$ for x^n, and again expanding in series about t^n, we obtain

$$\begin{aligned} x(t^{n+1}) = x(t^n) + \Delta t\, v^n + \frac{\Delta t^2}{2} a(x(t^n), t^n) \\ + \frac{\Delta t^3}{2} \dot{a}(x(t^n), t^n) \\ + \frac{\Delta t^4}{4} \{\ddot{a}(x(t^n), t^n) - \partial_x a(x(t^n), t^n) \\ \times a(x(t^n), t^n)\} + O(\Delta t^5) - d^{n+1}. \end{aligned} \tag{42}$$

If we use just the right value for v^n, we preserve Eq. (41). However, if $v^n = v(t^n)$ is used we obtain

$$d^{n+1} = -\frac{\Delta t^3}{3} \dot{a}(t^n, x(t^n)) + O(\Delta t^4), \tag{43}$$

which only differs from the error obtained from the leapfrog scheme Eq. (21), by a factor of two. Therefore, the same timestep control criteria derived in Section 5.1 (Eqs. (22)–(25)) still apply for the D1 scheme.

ACKNOWLEDGMENTS

This work was supported at University of California, Berkeley by the U.S. Department of Energy under Contract DE-FG03-86ER53220 and the U.S. Office of Naval Research under Contract N00014-85-K-0809. The work at Lawrence Livermore National Laboratory was performed under auspices of the U.S. DOE under Contract W-7405-ENG-48.

REFERENCES

1. A. Friedman, S. E. Parker, S. L. Ray, and C. K. Birdsall, *J. Comput. Phys.* **96**, 54 (1991).
2. A. Friedman, A. B. Langdon, and B. I. Cohen, *Commun. Plasma Phys. Controlled Fusion* **6**, 225 (1981).
3. A. B. Langdon and D. C. Barnes, in *Multiple Time Scales*, edited by J. U. Brackbill and B. I. Cohen, Computational Techniques (Academic Press, New York, 1985).
4. A. B. Langdon, B. I. Cohen, and A. Friedman, *J. Comput. Phys.* **51**, 107 (1983).
5. B. I. Cohen, A. B. Langdon, and A. Friedman, *J. Comput. Phys.* **46**, 15 (1982).
6. B. I. Cohen, A. B. Langdon, D. W. Hewett, and R. J. Procassini, *J. Comput. Phys.* **81**, 151 (1989).
7. C. K. Birdsall and A. B. Langdon, *Plasma Physics via Computer Simulation* (McGraw–Hill, New York, 1985).
8. W. S. Lawson, *J. Comput. Phys.* **80**, 253 (1989).
9. S. E. Parker, Ph.D. thesis, University of California, Berkeley, March 1990.
10. C. W. Gear, *Numerical Initial Value Problems in Ordinary Differential Equations*, (Prentice-Hall, Englewood Cliffs, NJ, 1971).
11. M. Keilhacker, G. Becker, *et al.*, *Plasma Phys. Controlled Fusion* **26**, 49 (1984).
12. R. J. Mason, *Phys. Fluids* **13**, 1042 (1970).
13. R. J. Mason, *Phys. Fluids* **14**, 1943 (1971).
14. I. Langmuir, *Phys. Rev.* **33**, 954 (1929).
15. G. A. Emmert, R. M. Wieland, A. T. Mense, and J. N. Davidson, *Phys. Fluids* **23**, 803 (1980).
16. D. A. Tidman and N. A. Krall, *Shock Waves in Collisionless Plasmas* (Wiley-Interscience, New York, 1971), p. 103.
17. S. S. Moiseev and R. Z. Sagdeev, *J. Nucl. Energy C* **43** (1963).
18. J. Ambrosiano, S. T. Brandon, and R. Lohner, in *Proceedings, 14th Int. Conference on the Numerical Simulation of Plasma, 0W2, Annapolis, MD 1991.*
19. S. E. Parker and C. K. Birdsall, *J. Comput. Phys.* **97**, 91 (1991).
20. D. W. Hewett and A. B. Langdon, *J. Comput. Phys.* **72**, 121 (1987).